

COMPUTER  
TECHNOLOGY

1620 GENERAL PROGRAM LIBRARY

Plot Subroutine

13.0.001

Disclaimer

Although this program was tested by its author prior to submission, no warranty, express or implied, is made by the author, 1620 USERS Group, or IBM as to the accuracy and functioning of the program and related program material and no responsibility is assumed by the author, 1620 USERS Group, or IBM in connection therewith.

1620 USERS GROUP PROGRAM REVIEW AND EVALUATION

(fill out in typewriter or pencil, do not use ink)

Program No. \_\_\_\_\_

Date \_\_\_\_\_

Program Name: \_\_\_\_\_

1. Does the abstract adequately describe what the program is and what it does? Yes        No         
Comment \_\_\_\_\_
2. Does the program do what the abstract says? Yes        No         
Comment \_\_\_\_\_
3. Is the Description clear, understandable, and adequate? Yes        No         
Comment \_\_\_\_\_
4. Are the Operating Instructions understandable and in sufficient detail? Yes        No         
Comment \_\_\_\_\_  
Are the Sense Switch options adequately described (if applicable)? Yes        No         
Are the mnemonic labels identified or sufficiently understandable? Yes        No         
Comment \_\_\_\_\_
5. Does the source program compile satisfactorily (if applicable)? Yes        No         
Comment \_\_\_\_\_
6. Does the object program run satisfactorily? Yes        No         
Comment \_\_\_\_\_
7. Number of test cases run \_\_\_\_\_. Are any restrictions as to data, size, range, etc. covered adequately in description? Yes        No         
Comment \_\_\_\_\_
8. Does the Program Meet the minimal standards of the 1620 Users Group? Yes        No         
Comment \_\_\_\_\_
9. Were all necessary parts of the program received? Yes        No         
Comment \_\_\_\_\_
10. Please list on the back any suggestions to improve the usefulness of the program.  
These will be passed onto the author for his consideration.

Please return to:

Mr. Richard L. Pratt  
Data Corporation  
7500 Old Xenia Pike  
Dayton, Ohio 45432

Your Name \_\_\_\_\_

Company \_\_\_\_\_

Address \_\_\_\_\_

User Group Code \_\_\_\_\_

THIS REVIEW FORM IS PART OF THE 1620 USER GROUP ORGANIZATION'S PROGRAM REVIEW AND EVALUATION PROCEDURE. NONMEMBERS ARE CORDIALLY INVITED TO PARTICIPATE IN THIS EVALUATION.

O

O

O

PLOT

SUBROUTINE

FOR THE IBM 1620 FORMAT FORTRAN

PROCESSOR

TABLE OF CONTENTS

Page

|                                     |    |
|-------------------------------------|----|
| Description of Subroutine . . . . . | 1  |
| Output . . . . .                    | 1  |
| Flow Chart . . . . .                | 3  |
| Sample Problem . . . . .            | 4  |
| Program Listing . . . . .           | 17 |

Jesse H. Poore, Jr.  
Computing Center  
University of Kentucky  
Lexington, Kentucky  
March, 1962  
Library Code **3143**

Modifications or revisions to this program, as they occur,  
will be announced in the appropriate Catalog of Programs  
for IBM Data Processing Systems. When such an announce-  
ment occurs, users should order a complete new program  
from the Program Information Department.

//

/\

PLOT  
PLOT SUBROUTINE FOR FORMAT FORTRAN

Purpose: This subroutine plots up to 9 curves simultaneously.

Form of Statement:

V = PLOT (FLT)  
where: V is a dummy variable name, which will be given the value FLT  
PLOT is the call name of the functional subroutine  
FLT is a floating point number which is the argument of the subroutine  
FLT must be scaled such that  
1.  $\leq$  FLT  $\leq$  80.

Description of Subroutine:

Only the integral part of FLT will be considered by this subroutine. The first time PLOT is entered, the digit "1" will be stored for punching in column FLT of the card (integral part of FLT). The second time PLOT is entered the digit "2" will be stored in column FLT of the card. If the argument of PLOT should happen to be the same for two different curves, the digit representing the latter curve will be plotted.

This procedure is repeated until:

- 1) FLT is zero, in which case the card will be punched as arranged by the previous entrances. A zero argument will clear the punch area to blanks and ready the subroutine for the next set of points. A zero argument may be the tenth entry, and is the only exception to the following paragraph.
- 2) A tenth entry, which is forbidden except for zero argument is attempted. In this event the message "PLOT" is typed to signal the operator that an attempt at plotting 10 curves has been fruitlessly attempted. The subroutine initializes and as a consequence of violating the restrictions, the first nine curves are lost.

If the argument (FLT) is out of range, i.e. if  $FLT < 1.$  or if  $FLT > 80.$ , the argument is typed, the typewriter spaces once, and the entrance attempt number is typed, which identifies the curve. The sequence of plots is not disrupted by such an occurrence.

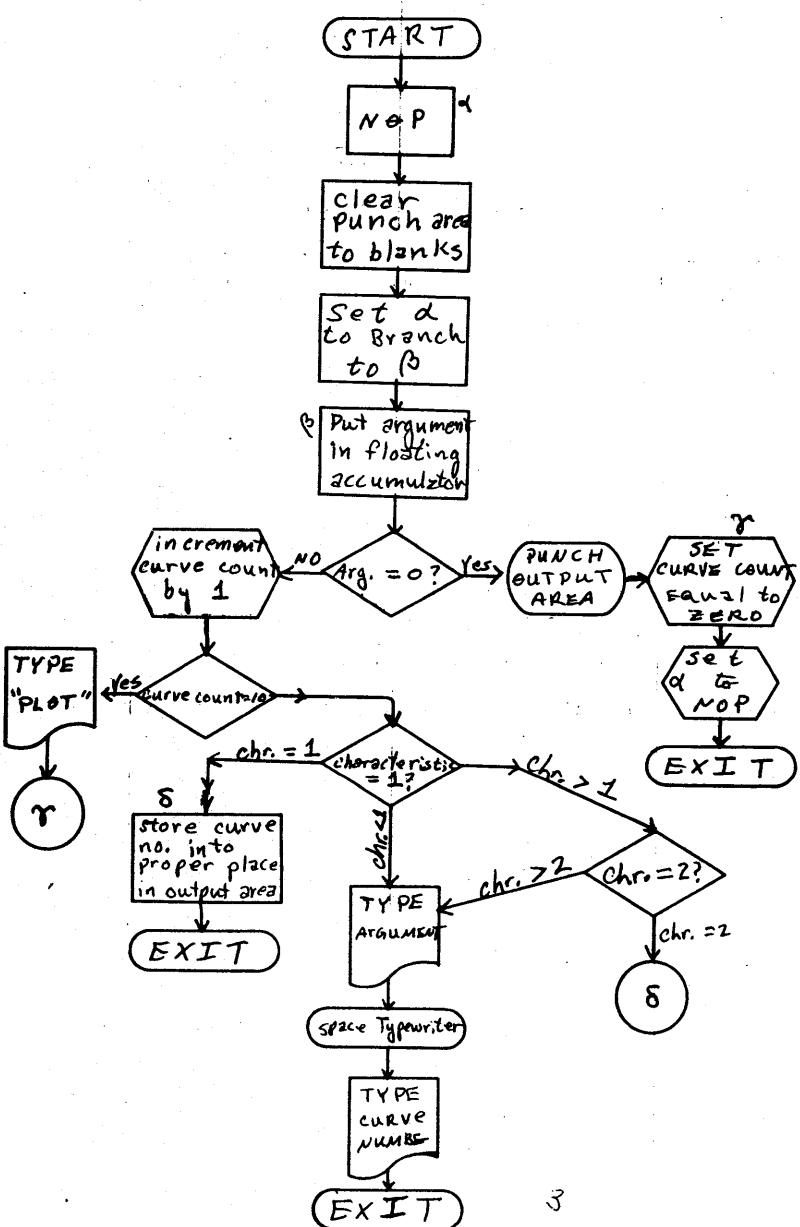
The output of the sample problems included herein was listed on an IBM 407 with an 80-80 control panel.

In the listing of the source program, the statement labeled L1 and the three following statements must be adjusted to agree with the subroutine number for various compiler decks.

Storage Requirement: 550 digits

Reminder: A functional subroutine must be preceded by some arithmetic statement before it is called.

## FLOW CHART



```

Z=1.
W=.05
X=0.

1  Y=PLOT(SIN(X)*39.+40.)
Y=PLOT(COS(X)*39.+40.)
Y=PLOT(SIN(X)/COS(X)+1.)
Y=PLOT(LOG(W)+4.)
Y=PLOT(EXP(X))
Y=PLOT(Z)
Y=PLOT((EXP(X)-EXP(-X))/2.+1.)
Y=PLOT(Z**(.6./5.))
Y=PLOT(13.+.301*X+.99876*X*X+.0076899*X*X*X)
Y=PLOT(0.0)
X=X+.03491
Z=Z+.5
W=W+1.
GO TO 1
STOP
END

```

|     |     |    |    |
|-----|-----|----|----|
| 8   | 9   | 1  | 2  |
| 8   | 4   | 1  | 2  |
| 78  | 4   | 1  | 2  |
| 768 | 4   | 1  | 2  |
| 7   | 8   | 4  | 2  |
| 7   | 684 | 9  | 2  |
| 7   | 68  | 9  | 2  |
| 7   | 648 | 9  | 2  |
| 7   | 68  | 9  | 2  |
| 7   | 648 | 9  | 2  |
| 7   | 6   | 8  | 2  |
| 7   | 6   | 8  | 2  |
| 7   | 46  | 8  | 2  |
| 7   | 46  | 8  | 2  |
| 7   | 4   | 6  | 89 |
| 7   | 4   | 6  | 9  |
| 7   | 4   | 6  | 9  |
| 7   | 4   | 6  | 98 |
| 7   | 4   | 6  | 9  |
| 7   | 4   | 6  | 8  |
| 75  | 4   | 6  | 9  |
| 75  | 4   | 6  | 9  |
| 75  | 4   | 69 | 8  |
| 75  | 4   | 69 | 8  |
| 75  | 4   | 9  | 8  |
| 75  | 4   | 69 | 8  |
| 7   | 4   | 9  | 8  |
| 7   | 4   | 9  | 8  |
| 7   | 4   | 96 | 8  |
| .7  | 4   | 96 | 8  |
| 7   | 4   | 9  | 6  |
| 7   | 4   | 9  | 6  |
| 75  | 4   | 9  | 6  |
| 75  | 4   | 9  | 6  |
| 75  | 4   | 9  | 6  |
| 75  | 4   | 9  | 6  |
| 753 | 4   | 9  | 6  |





```

X=.000318
Y=PLOT(40.)
Y=PLOT(X*SIN(1./X)*300.+40.)
Y=PLOT(0.0)
1 X=X+.000318
 GO TO 2
 STOP
END

```

6

10

The image shows a large grid of binary digits (0s and 1s) arranged in a specific pattern. The pattern consists of several vertical columns of 0s and 1s, with some columns containing only 0s or only 1s, and others containing both. The sequence of digits forms a recognizable word at the bottom. The word 'COMPUTER' is written in a bold, sans-serif font, with each letter composed of a unique arrangement of binary digits. The letters are positioned at the bottom of the grid, with 'C' on the left, followed by 'O', 'M', 'U', 'P', 'T', 'E', and 'R' on the right. The background of the grid is white, and the digits are black.

# **COMPUTER TECHNOLOGY**



16

\*\* FORTRAN PLOT SUBROUTINE  
 DORG 05000  
 START NOP L1,,0 05000 41 05096 00000  
 TFM L3+6,OUTPUT+1,017 05012 16 05042 05458  
 L2 TFM \*+9,80,010 05024 16 05033 00080  
 L3 TDM OUTPUT+1,,02 05036 15 05458 00000  
 J33 1,\* 05047 00001 @  
 AM L3+6,1,010 05048 11 05042 00001  
 SM L2+9,1,010 05060 12 05033 00001  
 BNE L3,,0 05072 47 05036 01200  
 TDM START+1,9,0 05084 15 05001 00009  
 L1 TF 60,19809,11 05096 26 00060 19809  
 SM 19809,2,10 05108 12 19809 00002  
 TF 58,19809,11 05120 26 00058 19809  
 AM 19809,2,10 05132 11 19809 00002  
 BD DIGIT,51,0 05144 43 05194 00051  
 WNCD OUTPUT+1,,0 05156 38 05458 00400  
 L5 TFM PLOT,,010 05168 16 05202 00000  
 TDM START+1,1,0 05180 15 05001 00001  
 BB 05192 42 00000 00000  
 DORG \*-9 05194  
 DIGIT AM PLOT,1,010 05194 11 05202 00001  
 CM PLOT,10,010 05206 14 05202 00010  
 BE PLT,,0 05218 46 05434 01200  
 L6 TFI L4+6,OUTPUT,017 05230 16 05376 05457  
 CM 60,1,10 05242 14 00060 00001  
 BE ONEDIG,,0 05254 46 05346 01200  
 BL RANGE,,0 05266 47 05384 01300  
 CM 60,2,10 05278 14 00060 00002  
 BH RANGE,,0 05290 46 05384 01100  
 CM 52,80,10 05302 14 00052 00080  
 BH RANGE,,0 05314 46 05384 01100  
 A L4+6,52,0 05326 21 05376 00052  
 B L4,,0 05338 49 05370 00000  
 DORG \*-3 05346  
 ONEDIG TD \*+23,51,0 05346 25 05369 00051  
 SM L4+6,,010 05358 12 05376 00000  
 L4 TD ,PLOT,1 05370 25 00000 05202  
 BB 05382 42 00000 00000  
 DORG \*-9 05384  
 RANGE TD 61,401 05384 25 00061 00401  
 WNTY 51 05396 38 00051 00100  
 SPTY 05408 34 00000 00101  
 WNTY PLOT,,0 05420 38 05202 00100  
 BB 05432 42 00000 00000  
 DORG \*-9 05434  
 PLT WATY PLTX,,0 05434 39 05539 00100  
 B L5,,0 05446 49 05168 00000  
 PUTOUT DS 80 05537 00080  
 OUTPUT DS 1,PUTOUT-80 05457 00001  
 PLOT DS 2,DIGIT+8 05202 00002  
 DC 1,@,DIGIT+9 05203 00001 #  
 PLTX DAC 5,PLOT 05539 00005X2 PLOT  
 DEND 00000

END OF PASSII